Complex & Intelligent Systems (2024) 10:1703–1714
https://doi.org/10.1007/s40747-023-01235-9

ORIGINAL ARTICLE

Improving table detection for document images using boundary

Yingli Liu[1,2] · Jianfeng Zheng[1,2] · Guangtao Zhang[1,2] · Tao Shen[1,2]

Received: 13 April 2023 / Accepted: 3 September 2023 / Published online: 30 September 2023

Abstract
Locating tables in document images is the first step to extracting table information, and high location precision is required. The dominant approach of table detection is based on an object detection algorithm, and the detector defines the prediction task as a regression problem, which inevitably leads to positioning errors. To address this issue, this paper presents an approach called Border Line Correction (BLC) to refine the rough prediction results of the original detector through the table boundary lines extracted from the document image. Our approach transforms the regression task into a classification problem, thus avoiding the inherent regression error of the object detection algorithm. Traditional annotation methods are inadequate for table detection tasks as they fail to capture the completeness and purity of the detection results. Therefore, this study treats the correct position of a table as a tolerance region. Additionally, to overcome the limitations of existing datasets in the materials domain, we collected 1183 samples from scientific literature in the materials field and created the MatTab dataset, annotating the tables with tolerance regions. This paper use Cascade RCNN with Swin Transformer as baseline models, and BLC is utilized to optimize the detection results. Experimental results demonstrate significant improvements with BLC at an IOU of 0.95 on the MatTab, ICDAR2019, and ICDAR2017 datasets. In MatTab, the percentage of correctly detected complete and pure tables increased from 72.3% to 82.1%.

Keywords Table detection · Deep learning · Document image · Object detection

Introduction

With the development of technology, a large number of electronic documents are generated in the production and scientific research activities in modern society. These documents are usually in unstructured forms, such as PDF, which is not conducive to retrieving required information. To efficiently exploit the data in these documents, it is necessary to develop an algorithm that can automatically analyze documents and accurately recognize the required data. The electronic document is composed of tables, illustrations, figures, text, and other elements. Due to the high information density of tables, extracting table content has become an important research direction in document analysis. In scientific literature, tables are commonly used to present experiment processes and results. Locating the tables and converting them into a structured format is particularly important. This conversion enables easier analysis of the data in subsequent stages.

The goal of table detection is to predict the position of the table in the document image, which is the upstream task of table interpretation. In the past few years, researchers use rule-based methods to separate tables from other elements [1–3], but the complex document layout and diversified tabular make the traditional method suffer from poor universality.

With the development of computer science and artificial intelligence, the concept of reinforcement learning and deep learning begins to attract attention in the field of machine learning and demonstrate promising results in various applications [4]. In particular, deep learning has a significant

✉ Tao Shen
shentao@kust.edu.cn

Yingli Liu
lyl@kust.edu.cn

Jianfeng Zheng
20202204306@stu.kust.edu.cn

Guangtao Zhang
20212204323@stu.kust.edu.cn

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Wujiaying Street, Kunming 650500, Yunnan, China

[2] Yunnan Key Laboratory of Computer Technologies Application, Kunming University of Science and Technology, Wujiaying Street, Kunming 650500, Yunnan, China

impact on image-based table detection methods. Under this influence, object detection methods are widely used. Obtaining feature maps of images is the basis of object detection, and appropriate features greatly affect the performance of the detection algorithm. Therefore, some studies focus on how to learn more significant table features from document images [5–7], while others pay attention to improving the detector [8–10]. The deep learning method demonstrate good generalization ability in table detection tasks and achieve state-of-art performance.

Considering the integrity of the table content, the predicted coordinates must frame the table entirely. However, most previous work are based on object detection generates bounding boxes by regressing a large set of proposals or anchors. No matter how the model structure is modified, the bounding boxes obtained by regression will inevitably suffer from deviations, thus losing table text or confusing other elements. It is common to optimize the original results through multiobjective multitasking [11]. Based on observations, it is found that most tables have clear border lines to distinguish them from other elements in the document, this paper proposes a method that combines table outlines to fine-tune the bounding box. Our approach involves three steps: first, lines are extracted by a line detection algorithm, such as Hough Transform [12], CannyLines [13], etc., and such algorithms will generate many lines that may not represent the table boundary; more importantly, these methods recognize line segments through pixel boundaries, which means that the identified line segment coordinates have no regression error; so, the second step is to classify these lines using feature maps extracted by the backbone network of the object detector. This step filter useless lines and send the table boundaries to the next step; following the discovery of table boundaries, the third step is to rectify the bounding boxes according to the line coordinates identified in the previous process, this is how we transform the regression task into a classification problem.

At present, many table detection studies use IOU-based evaluation metrics as the evaluation standard, but the improvement of our method on the original model cannot be clearly reflected in IOU-based metrics. This is because IOU-based metrics are designed for object detection in natural scenes, and IOU calculates indicators based on Intersection Over Union thresholds, which represent the deviation between the ground truth and the predicted bounding box, but in table detection task, deviations can lead to the loss of table information or the detection of other elements. In document images, there are many blank areas, indicating that the ground truth of the table is a hollow rectangular range, not just a bounding box. Additionally, even if a bounding box has a large IOU value with the ground truth, loss of table element is still possible. Figure 1 shows the disadvantage of using IOU to evaluate the location accuracy: the predicted bounding boxes have a large IOU value in Fig. 1a, c, but one loses the left elements, and the other one is mixed with half of the table title; meanwhile, both Fig. 1b, d shows successfully detected tables with a low IOU. Since the use of IOU cannot evaluate table detection effect well, this paper labels a hollow rectangle called "tolerance area" as the ground truth of tables, and it is used as a supplement to table evaluation. In this way, the tolerance area can be labeled in MatTab.

The main contribution of our research is summarized as follows:

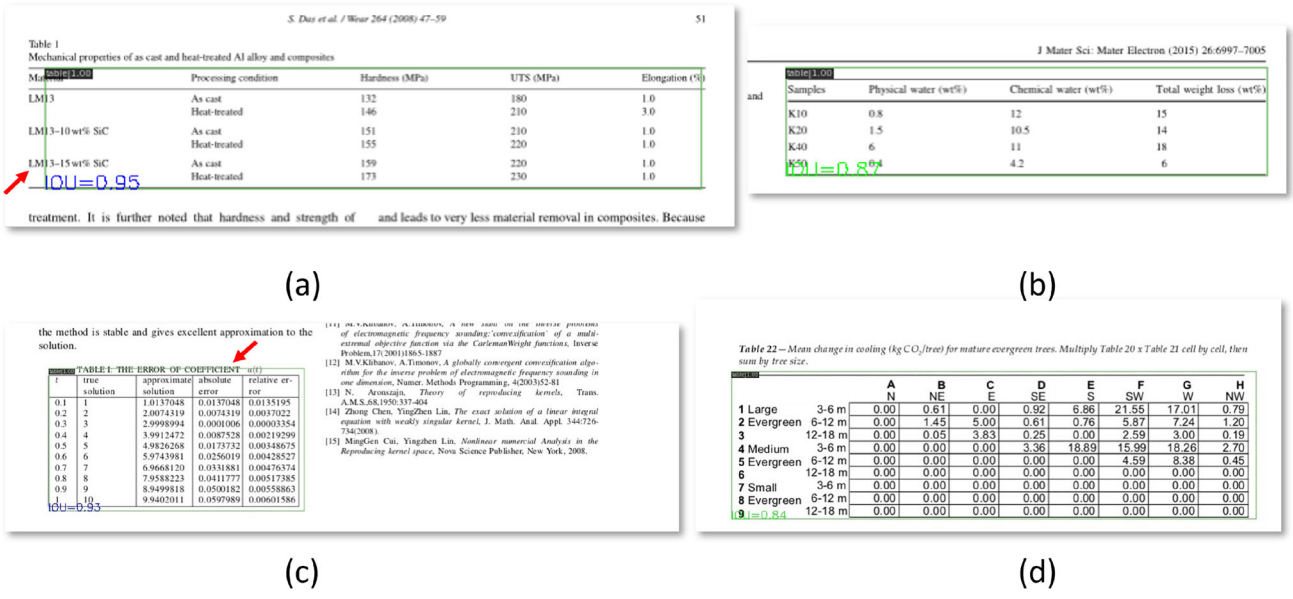

Fig. 1 Mask RCNN predicts result

- To avoid regression errors in typically object detection models, this paper presents an auxiliary module called Border Line Correction (BLC). BLC utilizes table boundary coordinates to refine the bounding boxes predicted by the original detector and transforms the regression task into a simple classification task, which helps to detect a complete and pure table. The BLC module is available at https://github.com/jeffrymain/table_detection_blc.git
- To overcome the defects of IOU-based metrics, a new ground truth is designed for the table detection task. Considering the characteristics of document images, the bounding box is replaced with a tolerance area as the ground truth of tabular, providing convenience for judging whether a detected table is complete and pure. Following the above principle, a new dataset called MatTab is provided. The dataset contains 1100 document images with tables, most of which are from scientific studies in the material domain.

Related work

Table detection

As early as 20 years ago, extracting table contents from documents has become a challenging problem. Most previous studies use a rule-based system to solve the problem of table location. However, due to the lack of generalization of rule-based methods and poor performance in complex layout documents, researchers began to use machine learning to locate tables. As deep learning achieving good results in object detection, the table positioning method based on deep learning has become mainstream.

Rule-based table detection methods analyze the layout of documents and identify tables according to the lines or the arrangement rules of text blocks in documents. These traditional methods focus on designing rules for document layout analysis. During the early 1990s, a method is introduced by Katsuhiko [1] to identify the structure of tables in document images. This approach relies on analyzing the regular arrangement of each cell within the table. Meanwhile, Chandran and Kasturi [3] use the horizontal and vertical split table structure in the document. With the popularity of machine learning, Kieninger propose the T-Recs system [14, 15] for unsupervised learning of table detection. Kasar et al. [16] applied the SVM algorithm to table detection, and they classified the horizontal and vertical line segments in the article to detect the table position. Traditional methods have many limitations, and rule-based methods can only be used for specific form of tables, while machine learning needs to extract effective features manually. As we seen, those traditional method prefer to identify table cells first, and combine those table cells into a complete table after that.

Currently, deep learning has a great impact on table detection tasks, and many researchers have great contributions in applying object detection algorithms to the table analysis task. Some researchers attempt to extract more accurate table feature maps. Gilani et al. are the first to apply an object detection method to location tables in document images. They perform three distance transformations on the original image, took the results of the three transformations of an RGB image's three channels as input, and achieved good performance on the UNLV dataset [17]. Similarly, Arif et al. citeRN16 also use the transformed image as the input of Faster R-CNN [18]. Li et al. note that the feature extraction network prefers to use the outline of the table as the table feature, and they use the GAN [19] network for adversarial training, so that the feature extraction network can extract the common features of the missing line table and the whole line table. With the development of object detection models, many researchers attempt to apply advanced models to table detection. CDeC-Net [20] combines deformable convolution with cascade Mask R-CNN to locate tables. Besides, DeepDeSRT [21] and CasTabDetectoRS [22] are all based on this idea.

The strategy of using additional information to improve table detection accuracy is also emerging. Sun [23] combines corner information to refine the predicted bounding box. Though this approach improves table detection precision, there are regression errors in the correction results, because it uses the same model to implement table detection and corner detection. What's more, they overlook the performance at a high IoU threshold.

Kara proposes an End-to-End system [24] to discover tabular structures, and the system is aware of the possibility of missing content when applying an object detection method to locate the table region. Kara adopted the strict Hough transform [12] to find only straight and continuous lines, so this method can only detect a few table boundaries. Unlike BLC which can distinguish line category, this solution uses a tight bounding box that encompasses all the intersecting lines, and then, the table is extended to cover this tight-line bounding box. Due to the use of strict Hough transform, a few tables can detect intersecting lines.

Unlike other mainstream research focusing on improving the deep neural network architecture, our approach focuses on the recognition of table boundaries. We eliminate the errors generated during the regression task of deep learning algorithms by considering the coordinates of the boundaries. This method is integrated into the framework of the object detection algorithm, making it applicable to any modern object detection algorithm.

Evaluate table detection

IOU-based metrics are widely used in the research of common object detection (e.g., ICDAR 2019 Competition [25]), and it evaluates the prediction effect through IOU. However, IOU can only show the offset between two regions, so it is not comprehensive in table detection tasks. To facilitate subsequent analysis, the detected tables must be complete and pure. The concept of complete and pure tables is proposed by Silva [25], and used in the ICDAR 2013 table competition [26]. Specifically, "complete" means that a region includes all table cells, while "pure" means that it does not include any table cells that do not belong to this table. A correct table detection is both complete and pure. At the time when deep learning had not yet emerged, scholars generally use the traditional bottom–up method for the table analysis task. Therefore, "pure" and "complete" are in the sense of table cells, which are not applicable to the object detection method.

Methods

In recent years, the object detection method achieves great performance in many application fields. However, different from the images in a natural scene, the table detection task for document images has its own characteristics: it pays more attention to the purity and integrity of detected tables. Our BLC and the new evaluation method on MatTab are dedicated to detecting complete and pure tables.

BLC architecture

Current object detection algorithms are developed based on feature maps extracted from the original images. Our method also needs these feature maps to classify lines, so BLC can assist other detectors to predict more accurate bounding boxes. The overall flow of BLC is illustrated in Fig. 2. It contains four components: a backbone to extract feature representations, a boundary extractor to find table borders, an object detector to predict rough bounding boxes, and the use of these boundaries to refine bounding boxes.

Backbone and detector

To attain high accuracy results, we use a model that is made by the combination of two approaches. Cascade RCNN is originally proposed by Cai and Vasconcelos to solve the problem of high-quality detection. Their approach involves utilizing a multi-stage model to improve detection accuracy. Liu et al. propose a new vision transformer, called Swin Transformer (SwinT) to serves as a general-purpose backbone for computer vision. Our experiments show that the cascade multi-stage model with the Swin Transformer backbone yields the best results.

SwinT adopts a hierarchical design, which includes four stages. Each stage reduces the resolution of the input feature map and expand the receptive field like CNN. In our implementation, we combine four stage outputs of SwinT with Feature Pyramid Network(FPN) neck to generate image's feature map.

Cascade RCNN is an object detection architecture that is an improvement and extension of the classic Faster RCNN. It adopts a cascaded approach by progressively refining the accuracy of object detection through multiple detection stages.

The architecture of Cascade RCNN mainly consists of three stages: the coarse detection stage, the intermediate detection stage, and the fine-grained detection stage. Each stage has an independent classifier and regressor to further filter and adjust the target boxes at each stage.

The entire architecture operates in a cascaded manner, with finely tuned classifiers and regressors at each stage, resulting in higher accuracy and stability in the final detection results. By cascading multiple detection stages, Cascade RCNN can gradually improve its detection capability for small objects while maintaining a low false-positive rate and improving the accuracy of object localization. As shown in Fig. 3, the image is fed into SwinT and FPN to transform into feature maps. The Bbox Heads take regions of interest (RoIs) features as input and predict bounding boxes, and "B" in this figure means the bounding boxes predicted by the heads. The Mask Head predicts the masks for the objects and "S" means a segmentation mask.

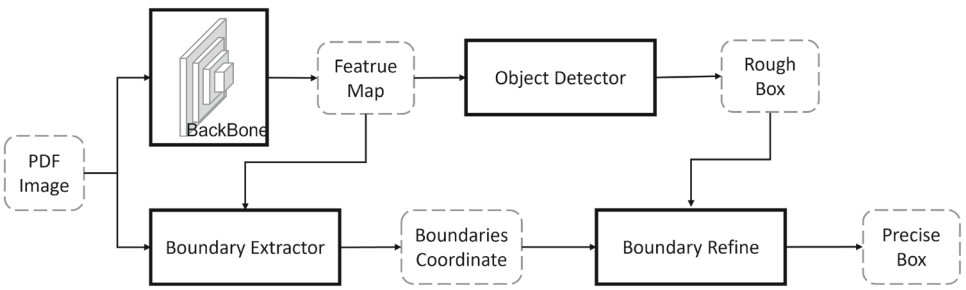

Fig. 2 BLC flow diagram

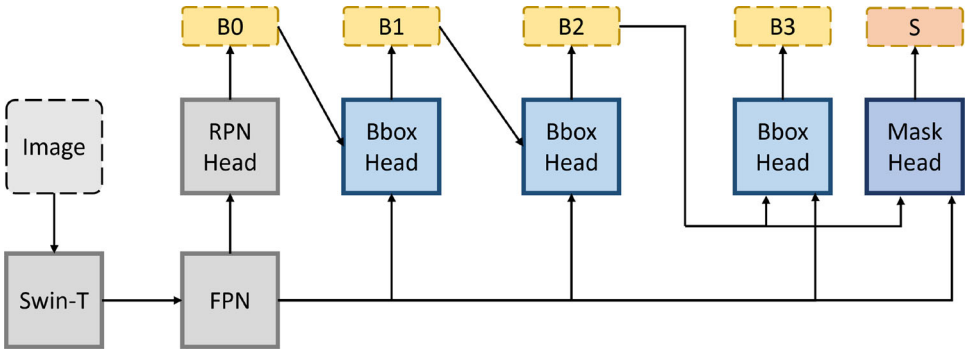

Fig. 3 Cascade RCNN architecture

Boundary extractor

In this module, all possible lines are obtained through the Line Extractor, and region proposals are generated for each line. Then, the proposals are projected on feature maps to obtain the line characters; finally, whether a line is the boundary of the table is judged according to its characters use Classify Head.

To generate region proposals for the line segment, the following rule is adopted: for a horizontal line, its proposal's width is equal to the line length, and the aspect ratio of the proposal is set to 4:1; for a vertical line, the proposal's high will be the line length, and its aspect ratio is changed to 1:4. RoiAlign converts these region proposals into fixed-size features to represent line features. This process is shown in Fig. 4. These line feature are then sent to classify the head to estimate whether this line segment is a table outline, and a simple full-connect neural network is used here as the classification head. Assuming each line feature represent with $A \in \mathbf{R}^{C\times W\times W}$, the classify score can be calculate by Eq. (1)

$$y = ReLU(W_a A), \tag{1}$$

where W_a is the fully connected neural network's weight parameters, W is a fixed number, we often set 7, and C is feature map's channel. In training, the ground truth is labeled for each line segment according to its distance to the nearest table boundary, and the loss is calculated with the cross-entropy loss function defined as Eq. (2)

$$L = \frac{1}{N}\sum_{i} -[y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)], \tag{2}$$

where y_i and p_i, respectively, denote the ground truth and the prediction probability, and N is the number of lines.

Boundary refine

The process of table boundary optimization can be divided into three parts: (1) match the correct table outline for the bounding box; (2) encode the bounding box and confuse the table outline coordinate; (3) decode and restore.

Match outline

In object detection task, a bounding box is usually represented using its corner coordinates, which can also represent with its four box boundaries. For each box boundary, the

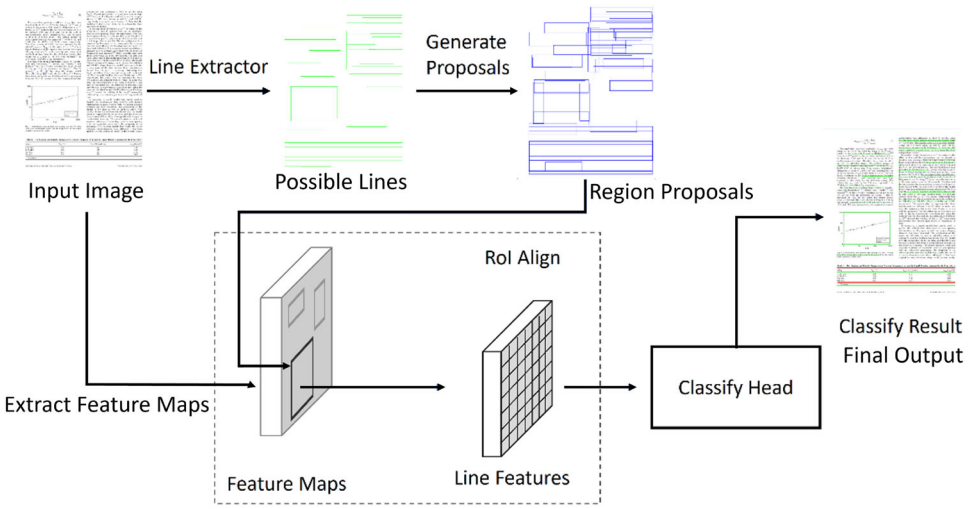

Fig. 4 Boundary extractor architecture

table boundary is classified by Boundary Extractor needs to be searched within the limited range: if an outline exists, this box boundary is said to match a table boundary. By matching table boundary for box boundary, we can get a match matrix $M \in \mathbf{R}^{4\times4}$ using those matched table boundaries, which as defined as Eqs. (3–4)

$$M = \begin{bmatrix} L_t & L_r & L_b & L_l \end{bmatrix} \tag{3}$$

$$L_i = \begin{cases} T_i, & match\ target \\ B_i, & otherwise, \end{cases} \tag{4}$$

where $T \in \mathbf{R}^4$ and $B \in \mathbf{R}^4$ mean matched table boundary and box boundary, respectively, and subscripts t, r, b, l represent four direction of bounding box, $i \in t, r, b, l$. Each boundary line is represented by the coordinates of two endpoints, since our research focuses on standard digital document images, and we assume that there are only horizontal and vertical line segments in the image; using the coordinates of left–right endpoint to represent horizontal boundaries and top–down to vertical.

Encode

Once we got the match matrix M, it should be encoded to M', and the element of M' can be calculated by Eq. (5)

$$M'_{i,j} = \begin{cases} y_c - M_{i+1,j}, & i = 1, 3 \\ x_c - M_{(i+1)mod4,j}, & i = 2, 4 \\ 0, & j = (i+2)mod4, \end{cases} \tag{5}$$

where x_c and y_c are the coordinate of center point of bounding box, each column of M' represents three directions of distance to a boundary line, remaining one direction is 0, and it means that a boundary line can affect the bounding box in three directions.

Confuse

The matrix M' contains all correction information of matched table boundaries extracted by Boundary Extractor and can be confused to $V \in \mathbf{R}^4$ by Eq. (6)

$$v_i = \sum_{j=1}^{4} \frac{M'_{i,j}}{3}, \tag{6}$$

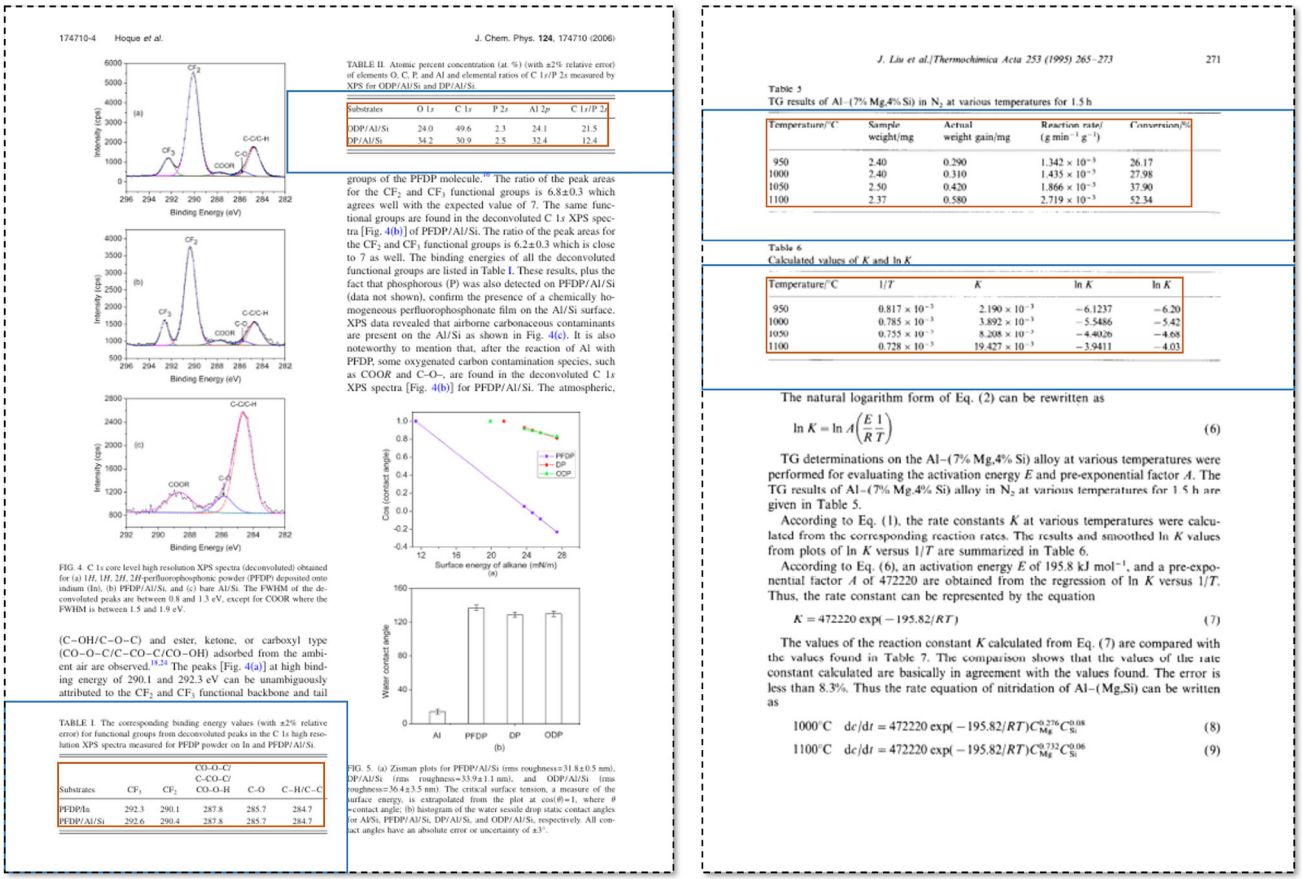

Fig. 5 MatTab example

where v_i is element of V, which represents the distance from center of rectified bounding box to its four boundary lines, and it can be easily reduced to coordinate form.

Experiments

In this section, we start by demonstrating a comparative analysis of various backbone models and different Line Extractor in Boundary Extractor. Then, we show the effectiveness of BLC by performing experiments with various object detector. And finally, we show the benchmarks of our model on public datasets. The experiment is performed on mmdetection deep learning framework with RTX 2060 GPU of 6 GB GPU memory, Intel(R) Core(TM) i5-9400F CPU @2.90 GHz and 16 GB of RAM. Training batch size is 1, and the input image is randomly rotated for data enhancement.

Data

Three datasets are used in this paper, namely ICDAR2019, ICDAR2017-POD, and MatTab.

ICDAR2019 is the most widely used dataset to evaluate table detection and table recognition. The entire dataset is divided into modern samples (electronic documents) and traditional samples (handwritten documents). The modern samples are obtained from different types of PDF documents, such as scientific journals and financial statements. The training set consists of 977 tables in 600 electronic document images; the test set consists of 449 tables in 240 images, and it has the largest proportion of tables without outlines in the three datasets.

ICDAR2017-POD is a common dataset for table detection structure identification, which consists of 549 training images and 243 test images. In addition to the table location, it also labels the text cells of the table. This dataset has more outline tables.

This paper also prepares the MatTab dataset to further evaluate the correctness of table detection and recognition. Experiment data in materials field are particularly important for the development of materials' genome initiative [27]. The MatTab dataset aims to address the scarcity of table detection dataset in the materials domain. We download 600 materials research papers using the keyword 'Al Si' from 'Web of Science' in PDF format. Following the resolution of the ICDAR 2019 dataset, we converted them into document images. Among these images, we selected 1183 images that contained tables, which served as samples for the MatTab dataset. We do not set any selection rules and processing; any document image that contained a table was included as a sample in the MatTab dataset. This ensured the authenticity of the samples and the diversity of the tables. The table positions in the images are manually annotated by three individuals, and the final positions were determined by taking the average values, thereby eliminating the subjective influence of the annotators.

With the availability of the MatTab dataset, researchers and practitioners in materials science can utilize it to advance methods for table extraction, analysis, and other related tasks. This dataset helps bridge the gap and enhances the capabilities of table extraction tools specifically tailored for the materials' science domain.

Table 1 Comparisons of three Line Extractor with different backbone network on MatTab

	TP	FP	FN	TN	P (%)	R (%)	Total
Hough [12]							
Swin	430	14	110	9334	96.84	79.62	9888
Resnet-50	373	14	167	9334	96.38	69.07	
MobileNetV2	204	8	336	9340	96.22	37.78	
FLD							
Swin	411	6	157	57413	98.56	72.36	57987
Resnet-50	135	6	433	57413	95.74	23.76	
MobileNetV2	33	1	535	57418	97.05	5.81	
CannyLines [13]							
Swin	379	14	138	46340	96.43	73.31	46871
Resnet-50	82	5	435	46349	94.25	15.86	
MobileNetV2	48	5	469	46349	90.57	9.28	

Evaluation metrics

For object detection, precision, recall, and score can be used for performance evaluation. Precision represents the proportion of correct detection results in the whole detected object. Recall is the proportion of true samples in all ground truth objects. Score considers both precision and recall, and it is calculated by taking the harmonic average of the two indicators.

The IOU value determines whether the detection is correct, which indicates the deviation between the predicted region and the ground truth. This method is used in ICDAR2019, with different IOU thresholds to measure location quality:

It is reasonable to use the IOU value in natural scenes, but as it is mentioned above, a more appropriate approach is to use the "tolerance region" to determine the correct detection. Thus, this approach is applied in MatTab.

The IOU-based metric only reflects the deviation between the bounding box and the ground truth, so it is difficult to judge whether the detected table is complete or pure. As shown in Fig. 1, even if the bounding box contains redundant content or misses detections, the IOU value is still high, and this situation is particularly serious in large tables; mean-

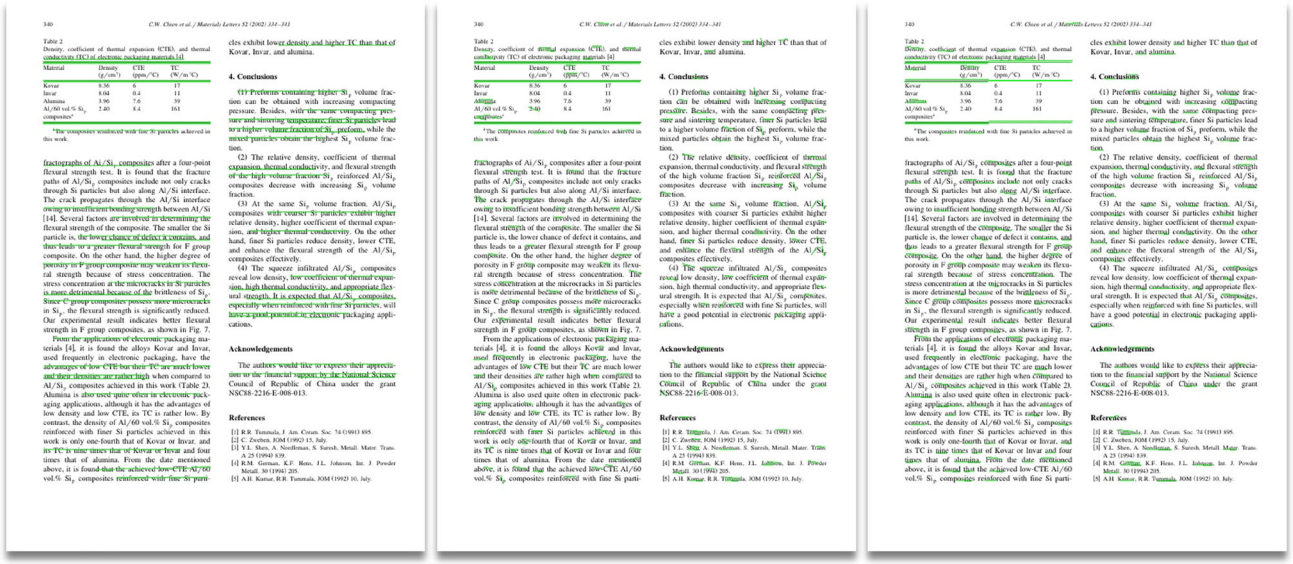

Fig. 6 Comparison of different line Extractor

while, a low IOU value does not affect the integrity and purity of tables.

Considering the characteristics of tables in the document, there is a lot of blank space between the table and other document elements. Any bounding box in these blank areas does not affect the correctness of the table, which indicates that a tolerance range is an exit for a bounding box. Therefore, it is believed that the ground truth of the table is a hollow rectangular area called the "tolerance area". Two ground truth bounding boxes are labeled to describe the tolerance area: mark a bounding box G^I with the smallest area without losing table information and mark another bounding box G^O with the largest bounding box without additional document elements. The label is shown in Fig. 5. These two bounding boxes describe the inner and outer boundaries of the table, and there are no other document elements within this region. If the detection results fall within these two boundaries, the table is considered pure and complete, and this is referred to as a correctly detection.

A correctly detected region is in the tolerance area, which is represented in Eq. (7), where T is a predicted table region

$$Detected = \begin{cases} True, & if G^I \subseteq T \subseteq G^O \\ False, & otherwise. \end{cases} \tag{7}$$

Experiment and analysis

We use different Line Extractor to extract all possible lines, and test the impact of various backbone on Boundary Extractor. We use three different Line Extractor Provided by the opencv, and the results are shown in Table 1. It can be found that the Boundary Extractor has good precision, which is a prerequisite for correcting table coordinates. Otherwise, excessive negative samples will have a negative effect during Boundary Refine process. Among three backbone network, the Transform architecture network SwinT is the most helpful for table boundary classification, which identify the largest number of table boundary. Among the three line Extractors with SwinT, 540 table boundaries are detected by Hough Transform algorithm, FLD with 568 boundaries, and CannyLines 517 boundaries. Though FLD detect the most table boundaries, but as shown in Fig. 6, FLD and CannyLines extract too much short lines, generate a large number of negative samples, and it greatly increases the computational complexity; therefore, the combination of Hough Transform and Swin is the most suitable choice in our Boundary Extractor.

Table 2 Results of Boundary Extractor in different dataset

	TP	FP	FN	TN	P (%)	R (%)	F1 (%)
ICDAR2019	542	110	284	10416	83.13	65.62	73.34
ICDAR2017	524	33	564	5511	94.08	48.16	63.71
MatTab	430	14	110	9334	96.84	79.62	87.39

We also test the classification performance of line extractor different datasets, the result is shown in Table 2, which shows high precision in MatTab and ICDAR2017, but recall is not ideal all datasets. In Boundary Extractor, we pay more attention to precision, because low recall rate just means that

Table 3 Influence of BLC on different detectors in MatTab

Model	BLC	0.8	0.9	0.95	C&P
BaiduOCR	×	0.91	0.76	0.38	0.489
BaiduOCR	✓	0.92	0.84	0.78	0.794
FasterRCNN	×	0.95	0.87	0.46	0.531
FasterRCNN	✓	0.95	0.89	0.81	0.747
MaskRCNN	×	0.96	0.91	0.64	0.626
MaskRCNN	✓	0.96	0.93	0.85	0.805
CascadeRCNN	×	0.97	0.95	0.83	0.763
CascadeRCNN	✓	0.97	0.94	0.89	0.821

BLC cannot process some tables, but wrong table boundary may lead to incorrect corrections.

Table 3 illustrates the influence of BLC on the F1-score of different object detection algorithms in MatTab dataset at high IOU threshold, last column C&P is complete and pure table evaluated by our 'tolerance area'. As seen in the table, F1-score significantly increases at IOU = 0.95, the increase trend is similar to C&P ratio.

Finally, we test the effect of BLC on other methods in IOU based metrics using public datasets, because they do not label tolerance area, and the results are reported in Tables 4 and 5. In both datasets, BLC demonstrate great effects on the benchmark model @IOU = 0.95.

To provide a more intuitive demonstration of the impact of BLCS on object detection models, we visualized the refine effects of BLC in our baseline model. As shown in Fig. 7, the left image in each visualization represents the predicted results of object detection. The middle image displays the localization coordinates after being corrected by BLCS. The right image shows the output of the line segment classification module, with green indicating line segments recognized by the model as table boundaries.

Table 5 BLC in ICDAR 2019 F1-scores

Model	BLC	IOU			
		0.6	0.8	0.9	0.95
BaiduOCR	×	0.791	0.649	0.514	0.395
BaiduOCR	✓	0.802	0.699	0.531	0.426
Li et al. [5]	×	0.0.924	0.891	0.785	0.536
Li et al.	✓	0.924	0.891	0.79	0.617
Kara et al. [24]	×	0.922	0.906	0.714	0.468
Kara et al.	✓	0.922	0.906	0.736	0.503
CascadeTabNet [10]	×	0.947	0.912	0.81	0.577
CascadeTabNet [10]	✓	0.947	0.912	0.824	0.751
Cascade RCNN	×	0.951	0.916	0.827	0.585
Cascade RCNN	×	0.951	0.916	0.837	0.753

Conclusions

To avoid the location error caused by the regression of an object detector in locating a complete and pure table, this paper proposes a BLC module to extract the line segment in document images and uses it to correct regression errors. BLC is an auxiliary method to enhance the detection accuracy of the original detector, and this method can be used by any modern detector. Since BLC shares the feature map with the original network, there is no need to add too many parameters, so the prediction speed is not affected. Also, this paper prepares a dataset called MatTab which labels the table's tolerance area to help check the integrity and purity of the table. Experiments indicate that BLC can assist other detectors to detect more complete and pure tables. Of course, our approach has limitations. The tables to be detected must have clear boundaries; otherwise, it will not be possible to optimize the rough results. Therefore, for borderless tables, we can only rely on the performance of the original object detection

Table 4 BLC in ICDAR 2017-POD F1-scores

Model	BLC	IOU			
		0.6	0.8	0.9	0.95
BaiduOCR	×	0.874	0.763	0.688	0.456
BaiduOCR	✓	0.874	0.772	0.72	0.52
Faster R-CNN+CCs+CRFs [28]	×	0.941	0.923	0.798	0.465
Faster R-NN+CCs+CRFs	✓	0.941	0.923	0.815	0.501
Faster RCNN+Corner Locating [23]	×	0.933	0.912	0.682	–0.413
Faster RCNN+Corner Locating	✓	0.933	0.912	0.698	0.462
Li et al [5]	×	0.939	0.927	0.743	0.422
Li et al	✓	0.939	0.928	0.761	0.463
Kara et al. [24]	×	0.941	0.935	0.771	0.457
Kara et al.	✓	0.941	0.935	0.803	0.498
Cascade RCNN	×	0.966	0.942	0.781	0.455
Cascade RCNN	✓	0.966	0.942	0.808	0.516

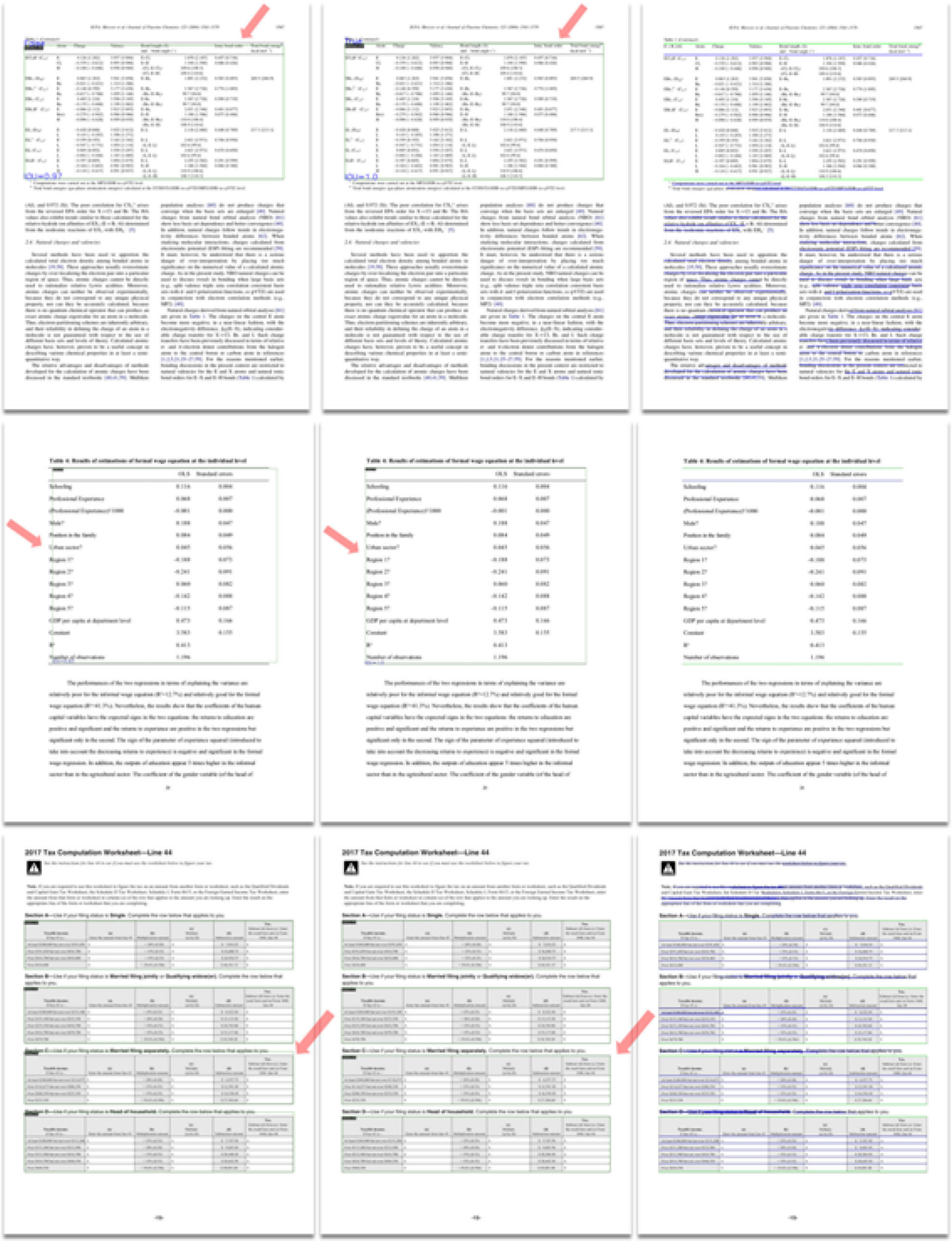

Fig. 7 Comparison of different line Extractor

algorithm. Additionally, although our Boundary Extractor achieved high accuracy, the recall rate was low, indicating that many tables do not participate in the optimization of BLC.

In the future, in addition to studying more advanced feature extraction algorithm and object detection algorithms, we will change the structure or training strategy of BLC to increase the recall rate to improve the impact of BLC on detection results. In this work, the tolerance area is only used for evaluation, so we will try to set this area as a training target in further studies. Furthermore, we will annotate table structure and content in the MatTab dataset. Building upon the work, we will investigate how to extract table structure and content.

Author Contributions YL: writing—original draft and writing—review & editing. JZ: methodology, software, data curation, writing—original draft, and writing—review & editing. GZ: software, data preparation, and writing—review & editing. TS: conceptualization, project administration, funding acquisition, and writing—review & editing.

Funding This work was supported by the National Natural Science Foundation of China (52061020, 61971208), Major Science and Technology Projects in Yunnan Province (202302AG050009), and Yunnan Fundamental Research Projects (202301AV070003).

Declarations

Conflicts of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Itonori K (1993) Table structure recognition based on textblock arrangement and ruled line position. In: Proceedings of 2nd international conference on document analysis and recognition (ICDAR '93), pp. 765–768. https://doi.org/10.1109/ICDAR.1993.395625
2. Kieninger T (1998) Table structure recognition based on robust block segmentation. In: DOCUMENT RECOGNITION V, vol. 3305, pp. 22–32. https://doi.org/10.1117/12.304642
3. Chandran S, Kasturi R Structural recognition of tabulated data. In: Proceedings of 2nd international conference on document analysis and recognition (ICDAR '93), pp. 516–519. https://doi.org/10.1109/ICDAR.1993.395683
4. Ottoni André LC, Nepomuceno MSdO Erivelton G (2022) Reinforcement learning for the traveling salesman problem with refueling. Complex Intell Syst 8:2001–2015. https://doi.org/10.1016/j.engappai.2020.103551
5. Li Y, Gao L, Tang Z, Yan Q, Huang Y (2019) A gan-based feature generator for table detection. In: 2019 International conference on document analysis and recognition (ICDAR), pp. 763–768. https://doi.org/10.1109/ICDAR.2019.00127
6. Gilani A, Qasim SR, Malik I, Shafait F (2017) Table detection using deep learning. In: 2017 14th IAPR International conference on document analysis and recognition (ICDAR), 01:771–776. https://doi.org/10.1109/ICDAR.2017.131
7. Casado-García A, Dominguez C, Heras J, Mata E, Pascual V (2020) Chapter 15. The Benefits of close-domain fine-tuning for table detection in document images, pp. 199–215. https://doi.org/10.1007/978-3-030-57058-3_15
8. Nazir D, Hashmi KA, Pagani A, Liwicki M, Stricker D, Afzal MZ (2021) Hybridtabnet: Towards better table detection in scanned document images. Appl Sci 11(18) https://doi.org/10.3390/app11188396
9. Nguyen D-D (2022) Tablesegnet: a fully convolutional network for table detection and segmentation in document images. Int J Document Anal Recognit (IJDAR) 25(1):1–14. https://doi.org/10.1007/s10032-021-00390-4
10. Prasad D, Gadpal A, Kapadni K, Visave M, Sultanpure K (2020) Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In: 2020 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), pp. 2439–2447. https://doi.org/10.1109/CVPRW50498.2020.00294
11. Dang Qianlong, GM, Gao Weifeng (2022) Multiobjective multitasking optimization assisted by multidirectional prediction method. Complex & Intell Syst 8:1663–1679. https://doi.org/10.1007/s40747-021-00624-2
12. Kiryati N, Eldar Y, Bruckstein AM (1991) A probabilistic hough transform. Pattern Recognit 24(4):303–316. https://doi.org/10.1016/0031-3203(91)90073-E
13. Lu X, Yao J, Li K, Li L Cannylines: A parameter-free line segment detector. In: 2015 IEEE international conference on image processing (ICIP), pp. 507–511. https://doi.org/10.1109/ICIP.2015.7350850
14. Kieninger T, Dengel A (1999) The T-Recs table recognition and analysis system, vol. 1655, pp. 255–269. Springer, Berlin
15. Kieninger T, Dengel A Applying the t-recs table recognition system to the business letter domain. In: Proceedings of Sixth International conference on document analysis and recognition, pp. 518–522. https://doi.org/10.1109/ICDAR.2001.953843
16. Kasar T, Barlas P, Adam S, Chatelain C, Paquet T (2013) Learning to detect tables in scanned document images using line information. In: 2013 12th international conference on document analysis and recognition, pp. 1185–1189. https://doi.org/10.1109/ICDAR.2013.240
17. Kieninger T, Dengel A (2005) An approach towards benchmarking of table structure recognition results. In: Eighth international conference on document analysis and recognition (ICDAR'05), pp. 1232–12362. https://doi.org/10.1109/ICDAR.2005.47
18. Ren S, He K, Girshick R, Sun J (2017) Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
19. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative Adversarial Nets

20. Agarwal M, Mondal A, Jawahar CV Cdec-net: Composite deformable cascade network for table detection in document images. In: 2020 25th international conference on pattern recognition (ICPR), pp. 9491–9498. https://doi.org/10.1109/ICPR48806.2021.9411922
21. Schreiber S, Agne S, Wolf I, Dengel A, Ahmed S (2017) Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR), 01:1162–1167. https://doi.org/10.1109/ICDAR.2017.192
22. Hashmi KA, Pagani A, Liwicki M, Stricker D, Afzal MZ (2021) Castabdetectors: Cascade network for table detection in document images with recursive feature pyramid and switchable atrous convolution. J Imag 7(10) https://doi.org/10.3390/jimaging7100214
23. Sun N, Zhu Y, Hu X (2019) Faster r-cnn based table detection combining corner locating. In: 2019 international conference on document analysis and recognition (ICDAR), pp. 1314–1319. https://doi.org/10.1109/ICDAR.2019.00212
24. Kara E, Traquair M, Simsek M, Kantarci B, Khan S (2020) Holistic design for deep learning-based discovery of tabular structures in datasheet images. Eng Appl Artificial Intell 90 https://doi.org/10.1016/j.engappai.2020.103551
25. Gao L, Huang Y, Déjean H, Meunier J-L, Yan Q, Fang Y, Kleber F, Lang E (2019) Icdar 2019 competition on table detection and recognition (ctdar). In: 2019 International conference on document analysis and recognition (ICDAR), pp. 1510–1515. https://doi.org/10.1109/ICDAR.2019.00243
26. Göbel M, Hassan T, Oro E, Orsi G (2013) Icdar 2013 table competition. In: 2013 12th international conference on document analysis and recognition, pp. 1449–1453. https://doi.org/10.1109/ICDAR.2013.292
27. Pablo MAW, Jackson Nicholas E (2019) New frontiers for the materials genome initiative. npj Comput Mater 5, 41 https://doi.org/10.1038/s41524-019-0173-4
28. Gao L, Yi X, Jiang Z, Hao L, Tang Z (2017) Icdar2017 competition on page object detection. In: 2017 14th IAPR International conference on document analysis and recognition (ICDAR), 01:1417–1422. https://doi.org/10.1109/ICDAR.2017.231